NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

OPTIMIZATION MODELS
FOR
SYNCHRONIZATION PLANNING

by

Christopher Morey

September 1991

Thesis Co-Advisor: David A. Dryer, CPT, USA
Thesis Co-Advisor: Siriphong Lawphongpanich

Approved for public release; distribution is unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL *(If applicable)* 55	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000	7b ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL *(If applicable)*	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS *(City, State, and ZIP Code)*	10 SOURCE OF FUNDING NUMBERS Program Element No / Project No / Task No / Work Unit Accession Number

11 TITLE *(Include Security Classification)*
Optimization Models For Synchronization Planning (UNCLASSIFIED)

12 PERSONAL AUTHOR(S) Christopher Morey

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT *(year, month, day)*	15 PAGE COUNT
Master's Thesis	From To	1991 September	72

16 SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS *(continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUBGROUP	Synchronization, Battlefield Operating Systems, Optimization, Linear Programming, Integer Programming, Network

19 ABSTRACT *(continue on reverse if necessary and identify by block number)*

Planning for the synchronization of activities on the battlefield for an Army battalion task force requires detailed planning for movement of subordinate units, allocation of personnel and assets to tasks, and many other activities to ensure that maximum damage is inflicted on an enemy force. Currently, this synchronization planning is done manually by task force staff officers, primarily the operations officer. The process is time consuming and most often results in a plan which is feasible, but not necessarily optimal. Two optimization models are developed to aid in the synchronization of task force activities. One of the models determines the feasibility of a course of action to aid the operations officer in developing a maneuver plan. The second model aids the engineer officer in allocating engineer assets to maximize the combat value of tasks. When implemented on computers, these models are flexible in that they allow for changes to be affected quickly. Hence, more alternatives can be considered in a short period.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Siriphong Lawphongpanich	22b TELEPHONE *(Include Area code)* (408) 646-2160 22c OFFICE SYMBOL 55Lp

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Approved for public release; distribution is unlimited.

Optimization Models
For
Synchronization Planning

by

Christopher Morey
Captain, United States Army
B.S., United States Military Academy, 1982

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1991

Peter Purdue, Chairman
Department of Operations Research

ABSTRACT

Planning for the synchronization of activities on the battlefield for an Army battalion task force requires detailed planning for movement of subordinate units, allocation of personnel and assets to tasks, and many other activities to ensure that maximum damage is inflicted on an enemy force. Currently, this synchronization planning is done manually by task force staff officers, primarily the operations officer. The process is time consuming and most often results in a plan which is feasible, but not necessarily optimal.

Two optimization models are developed to aid in the synchronization of task force activities. One of the models determines the feasibility of a course of action to aid the operations officer in developing a maneuver plan. The second model aids the engineer officer in allocating engineer assets to maximize the combat value of tasks. When implemented on computers, these models are flexible in that they allow for changes to be affected quickly. Hence, more alternatives can be considered in a short period.

TABLE OF CONTENTS

I. INTRODUCTION

One of the tenets of the Army's Air-Land Battle doctrine is synchronization which is "...the arrangement of battlefield activities in time, space, and purpose to produce maximum relative combat power at the decisive point." [Ref 1:p. 17] Put simply, synchronization is ensuring that the appropriate assets are in the appointed place at the right time. The primary goal of synchronization is to develop a plan which utilizes the capabilities of the available assets toward achieving the desired end result of inflicting the maximum amount of damage on an enemy force. This synchronization planning process commonly begins hours or days before the impending conflict on the battlefield, and includes the development of a unit's ground tactical plan, allocation of assets to tasks which must be performed, reconnaissance, rehearsals, movement of subordinate units, and resupply. In this study, two components of the planning process at battalion task force level are analyzed and studied. These components are the task force's maneuver plan and the allocation of engineer assets.

A. CURRENT SYNCHRONIZATION PLANNING PROCESS

The formal start of the planning process occurs when the task force receives its next operations order (mission) from

its parent brigade. [Ref 2:p. 4-8] Upon receipt of the operations order from the brigade, the task force intelligence officer develops likely enemy courses of action. Based on these potential enemy courses of action and with guidance from the commander, the operations officer develops several feasible courses of action (COA's) for the task force. Only one of the feasible COA's will be selected by the commander.

In the process of developing COA's, the operations officer tries to consider all possible engagement scenarios which are tactically sound. These engagement scenarios are then used to develop feasible courses of action. The feasibility is determined by, among other considerations, the ability of subordinate units, such as infantry and armor companies, to accomplish the required tasks and movements in the time frame deemed to be tactically advantageous. [Ref 3:p. E-10] Currently, the operations officer performs the necessary time-distance calculations by hand if the feasibility of a COA is not readily apparent. This process is generally time-consuming and prone to errors. Moreover, the lengthy calculations allow the operations officer to examine only a few scenarios since one COA must be selected and orders must be distributed in a relatively short period of time. The entire planning process, writing and distribution of orders to subordinates should only take one-third of the time available to the task force before the operation begins. If a mission

is to commence in twenty four hours, the order should be in the hands of the company commanders in eight hours.

One goal of this thesis is to develop a linear programming model to determine the feasibility of unit movements for a given course of action. When implemented on a microcomputer, the model would provide the necessary answer quickly, thereby allowing more COA's to be explored. This would then present the commander with more feasible COA's from which the most effective could be selected.

Once the commander decides on a COA, the operations officer and the remainder of the staff conduct <u>detailed</u> planning to synchronize the efforts of the task force assets and to ensure that all required tasks are accomplished prior to the commencement of battle. [Ref 3:p. 7-2] Part of the detailed planning involves planning the use of engineer assets such as bulldozers and engineer squads. This part of the plan is referred to as the <u>engineer plan</u> and is the responsibility of the engineer officer who has control of the assets.

In developing the engineer plan, the engineer officer is given a list of tasks and associated priorities. Ideally, one would expect the engineer officer to produce an engineer plan which maximizes the 'combat value' of the completed tasks. Considering the number of possible assignments of assets to tasks and possible sequences in which the tasks are to be performed, it would be unrealistic to expect the engineer

officer to manually produce a plan which maximizes the combat value.

To aid in the development of the engineer plan, this thesis also proposes a linear integer programming model to sequence and assign assets to tasks. When implemented, this model would help expedite the development of an optimal engineer plan and free up time that the engineer officer would have to spend manually devising a feasible engineer plan. Moreover, the model would also allow for any change in the final COA to be quickly incorporated into the engineer plan.

In summary, this thesis focuses on the development of two mathematical programming models: the Maneuver and the Engineer models. The first model determines the movement feasibility of a course of action. Given a feasible course of action, the Engineer model then generates an optimal schedule for engineer assets to perform the required tasks.

B. PRIOR WORK

The only known document which describes a systematic approach to synchronization planning is the masters thesis by Major C. Long. [Ref 4] The primary goal of his thesis is to develop a training tool called a synchronization planning matrix. This matrix consists of a single chart detailing the necessary steps in synchronizing the battlefield activities. The Army groups these activities into seven battlefield

operating systems (BOS). Major Long ordered these seven BOS into the following logical planning sequence: [Ref 4:p. 47]

1. Intelligence
2. Maneuver
3. Mobility/Countermobility/Survivability
4. Fire Support
5. Air Defense Artillery
6. Command and Control
7. Combat Service Support

Major Long's approach to synchronization is to manually plan for all of the activities in the BOS on one single chart. When plans become asynchronous, they would become apparent on the chart. This thesis builds upon Major Long's framework to synchronization by demonstrating that the activities in the first three BOS can be formulated as optimization models. Hence, they produce optimal plans and can be solved quickly on the computer. As for the remaining BOS, similar models can be constructed and are left for future investigation.

C. THESIS OUTLINE

Chapter II begins with a general overview of the planning process and describes the development of the courses of action. A sample maneuver plan is then used to illustrate the formulation of the Maneuver model.

Using the output from the Maneuver model, Chapter III discusses the development and formulation of the Engineer model which schedules engineer assets to perform tasks in the third BOS (Mobility/Countermobility/Survivability).

Chapter IV examines the results of the model as implemented for a maneuver plan devised with the help of a current battalion operations officer, MAJ William Odom, of the 3rd Battalion, 9th Infantry Regiment, 7th Infantry Division (Light). This chapter ends with analysis of the model and how common changes impact the output that the model generates.

Finally, Chapter V gives conclusions regarding the development of the models and the usefulness of the output. Recommendations are then given for further research efforts.

II. THE MANEUVER MODEL

In the actual planning of an upcoming battlefield operation, the operations and intelligence officers work closely to develop a plan to counter the probable enemy action. The planning process typically begins with the intelligence officer analyzing information regarding the enemy's movements and battle doctrine. After the intelligence officer's initial analysis of the enemy situation, the operations officer develops several alternate courses of action (COA's) for the task force. One of the major tasks for the operations officer is to determine which of the alternate COA's are operationally feasible. [Ref 3:p. E-10] The linear programming model developed below facilitates this task.

A. BACKGROUND

In essence, a COA consists of activities which must be performed in some logical order. In the model below, the activities and their logical order are represented as a network similar to those utilized in the Critical Path Method (CPM) or Project Evaluation and Review Technique (PERT). [Ref 5:p. 328] To illustrate the representation and define nomenclature, consider the sample COA depicted in Figure 2.1. In this figure, the task force is facing a Soviet style motorized rifle regiment. Based on the information from the

intelligence officer, the order of presentation of enemy forces begins with the regiment's advance guard, which comprises three smaller elements. The main body of the regiment follows the advance guard by twenty to thirty kilometers. [Ref 6:p. 5-32]

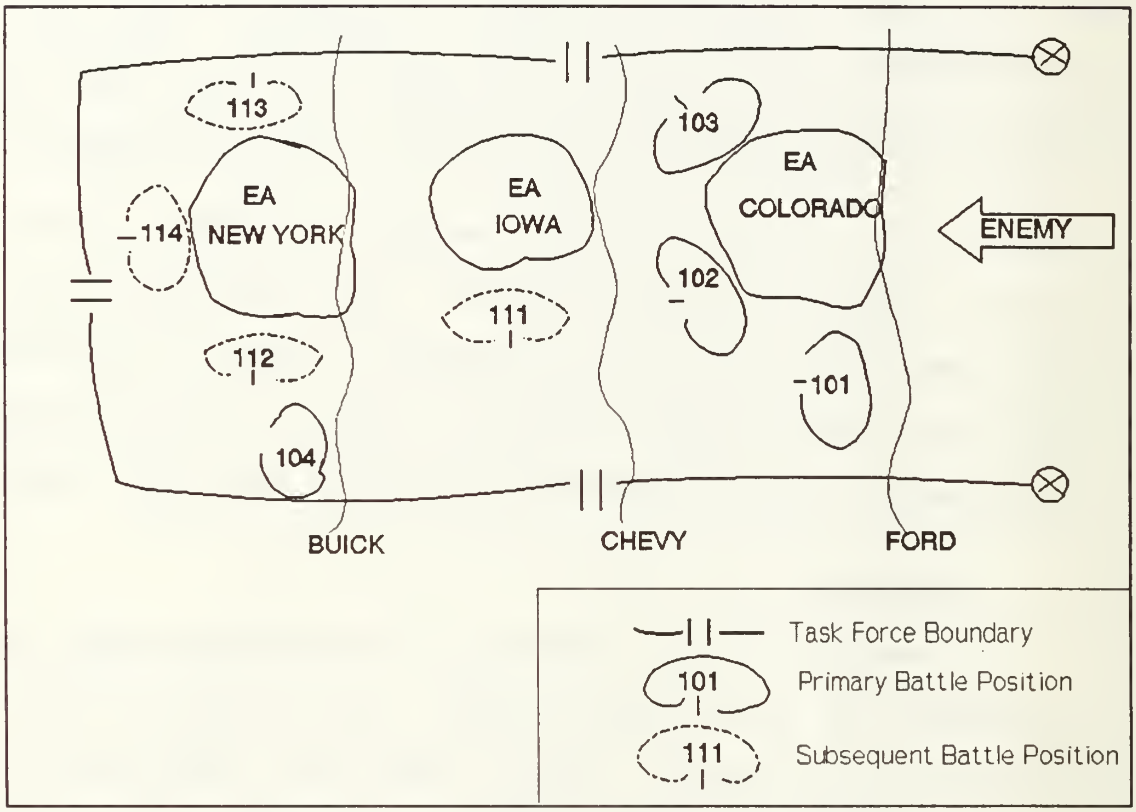

Figure 2.1 Sample Course of Action

The intelligence analysis indicates that the attack will occur in the northern half of the task force's sector, shown in Figure 2.1, since the terrain to the south is too restrictive for the movement of the enemy force. One COA calls for two tank companies to engage the advance guard from

battle positions (BP's) 102 and 103 in engagement area (EA) Colorado after the enemy crosses phase line (PL) Ford. This engagement will break off prior to the arrival of the enemy's main body. Once the initial engagement is broken off, the two tank companies will reposition to BP's 112 and 113 respectively. To engage the remnants of the advance guard in EA Iowa, a tank-heavy team (two tank platoons and one mechanized infantry platoon) will simultaneously move from its security position at BP 101 to BP 111. Once it is clear that the enemy is continuing its movement on the expected avenue, a mech-heavy team (one tank platoon and two mechanized infantry platoons) will reposition from their security position at BP 104 to BP 114. Finally, the enemy's main body will be defeated in EA New York after the enemy has crossed PL Buick.

Table 2.1 lists the activities with their precedence relationship. As an example, the first row shows that tank company 1 can begin to move from BP 102 after the enemy's main body is five kilometers from PL Ford. However, it must arrive at BP 112 prior to the arrival of the enemy's main body at PL Buick. In modeling these activities as a network, it is more convenient to convert the spatial information for the enemy in Table 1 to the temporal information in Table 2.2.

In Table 2.2, 'H' represents the time that the advance guard first enters the task force sector. The added minutes

Table 2.1: ACTIVITY PRECEDENCE
(Spatial Relationship)

Task Force Activity	Begin Activity After	Complete Activity Before
Move Tank Company 1 from BP 102 to BP 112	Enemy Main Body 5 Kilometers from PL Ford	Enemy Main Body at PL Buick
Move Tank Company 2 from BP 103 to BP 113	Enemy Main Body 5 Kilometers from PL Ford	Enemy Main Body at PL Buick
Move Tank Team from BP 101 to BP 111	Tank Companies Displaced from BP's 102 and 103	Lead Elements of Advance Guard at PL Chevy
Move Mech Team from BP 104 to BP 114	Enemy Main Body at PL Chevy	Enemy Main Body at PL Buick

are movement times which are estimated from the enemy's doctrinal movement rate, the terrain being traversed, weather conditions, and the impact of task force actions.

In developing plans to counter the enemy force, the operations officer uses the times displayed in Table 2.2 as a framework for the movement of task force units. Henceforth, the enemy activities and their associated times are referred to as 'critical activities' and 'critical times', respectively. Battlefield movements must therefore be planned in response to these activities and times.

Table 2.2: ACTIVITY PRECEDENCE
(Temporal Relationship)

Task Force Activity	Begin Activity	End Activity
Move Tank Company 1 from BP 102 to BP 112	H + 50 minutes	H + 120 minutes
Move Tank Company 2 from BP 103 to BP 113	H + 50 minutes	H + 120 minutes
Move Tank Team from BP 101 to BP 111	Tank Companies Displaced from BP's 102 and 103	H + 90 minutes
Move Mech Team from BP 104 to BP 114	H + 100 minutes	H + 120 minutes

Movement is one activity in the Maneuver battlefield operating system (BOS). Movement consists of the following three separate components:

1. Displace from the current position.
2. Move to a new position.
3. Occupy the new position.

Figure 2.2 depicts the movement components modeled in a network format.

In this figure, the nodes represent the start or the end of movement components by task force units. When the node is shaded, then it represents a critical activity and has an associated critical time. Associated with each arc, there is time to complete the corresponding movement component. Generally, combining the 'move to' and 'occupy' arcs

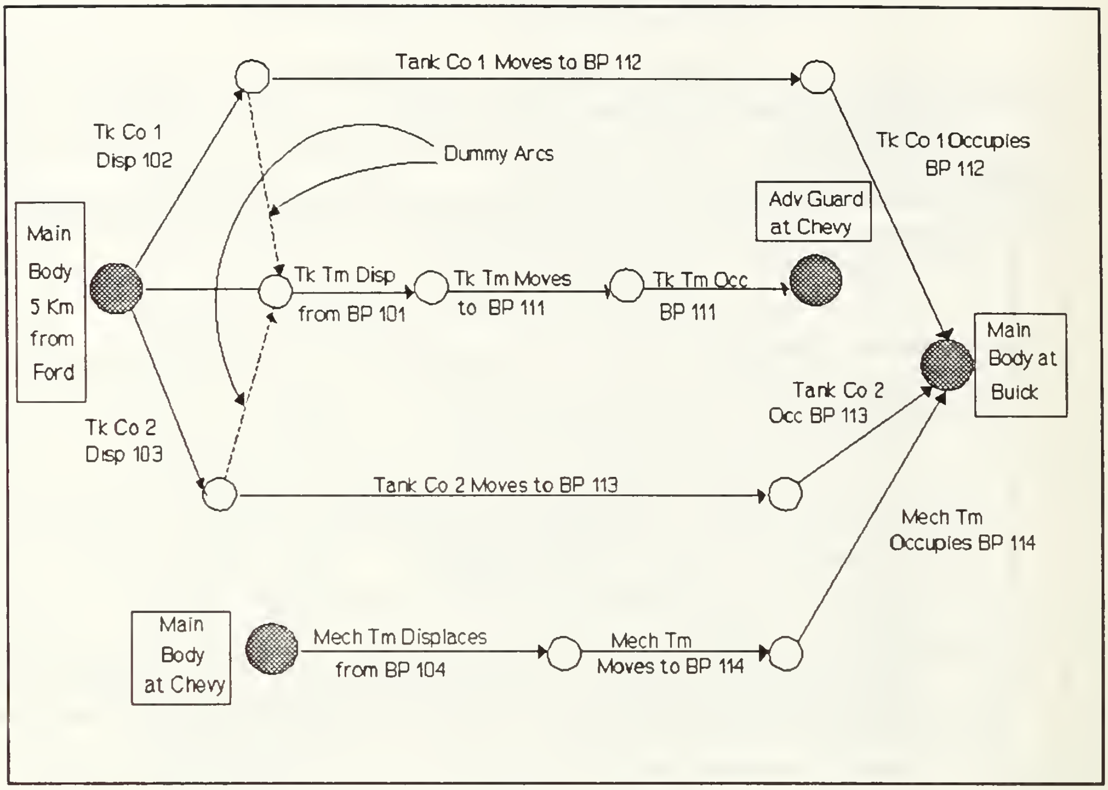

Figure 2.2 Movement Component Network

(components) into one arc representing the combined components will, in most cases, emulate actual occurrences on the battlefield since a unit usually occupies a position upon arriving. However, the 'move to' and 'occupy' components are separated to allow for a unit to move to the vicinity of the new position and remain in a concealed location to avoid early detection by the enemy. Upon receipt of a command to occupy, the unit will occupy the position to engage the enemy. Associated with each arc in the network is the time for the unit to complete the corresponding component of the task. These times are constant and are calculated from proposed or rehearsed rates of movement which depend on terrain, weather,

and time of day, etc. In the next subsection, a linear programming model which provides the solution for determining the earliest and the latest times to begin each movement component is described.

B. LINEAR PROGRAMMING FORMULATION

Below, the linear programming formulation to find the earliest and latest start times for each movement component is stated. This formulation includes two objectives: one to find the earliest start times and the other to find the latest.

Indices:

i,j desired task force activities to be performed, where $i,j=1,2,3,...,I$

k critical activities (for the enemy), where $k=1,2,3,...,K$

Given and Derived Data:

t_i time required to execute movement component i

s_k critical time for critical activity k

p_{ij} 1 if movement component i precedes movement component j; 0 otherwise

q_{ik} 1 if movement component i precedes critical activity k; 0 otherwise

r_{ki} 1 if Critical activity k precedes movement component i; 0 otherwise

Variables:

x_i start time of movement component i

z_i auxiliary variable representing additional time needed to complete movement component i

Formulation:

$$\textit{Minimize} \quad \sum_{i=1}^{I} X_i \qquad \textit{(To calculate earliest start time)}$$

$$\textit{Maximize} \quad \sum_{i=1}^{I} X_i \qquad \textit{(To calculate latest start time)}$$

Subject to:

$$x_i+t_i-x_j \quad \leq 0 \quad \forall \; i,j \quad \textit{such that } p_{ij}=1 \qquad (1)$$

$$x_i+t_i-z_i-x_k \quad \leq 0 \quad \forall \; i,k \quad \textit{such that } q_{ik}=1 \qquad (2)$$

$$x_i-s_k \quad \geq 0 \quad \forall \; k,i \quad \textit{such that } r_{ki}=1 \qquad (3)$$

$$x_i \geq 0, \; z_i \geq 0 \quad \forall \; i \qquad (4)$$

In constraint (1), the finish time of movement component i, i.e., $x_i + t_i$, must be less than the start time of the movement component j. In other words, constraint (1) ensures that movement component i precedes j. In constraint (2), if z_i equals zero, then the finish time of movement component i must be less than or equal to the start time of critical activity k. Constraint (3) simply requires that movement component i starts after critical activity k.

The variable z_i plays a central role in determining the feasibility of a COA. In the first objective, i.e., minimizing the sum of x_i, if the optimal value of z_i is positive for some i, then the COA is infeasible and the second

objective should not be considered. If the optimal value of z_i is zero for all i, then resolving the problem with the second objective, i.e., maximizing the sum of the x_i, while fixing all z_i to zero, would yield the latest start times.

Given a COA, the above formulation can determine whether the COA is feasible. If it is feasible, solving the linear programming problem twice, each time with a different objective, yields the earliest and latest start times for each movement component. The operations officer, at this point, would choose a start time within each feasible range to maximize the tactical advantage in defeating the enemy.

III. THE ENGINEER MODEL

To successfully carry out the selected COA as planned by the operations officer, the preparation of the battlefield and a variety of other activities must be completed prior to the start of the battle. Preparing the battlefield for a defense includes, among other things, digging vehicle positions, emplacing wire obstacles, and laying mines. Preparation for an attack or other offense missions includes clearing minefields and removing obstacles. These mobility, countermobility, and survivability tasks are the responsibility of the engineer officer who has control over all of the task force engineer assets, e.g. bulldozers, Armored Combat Earthmovers (ACE), and engineer squads.

The usual sequence of events which precedes the start of the engineers' work transpires as follows: [Ref 4:p. 5-2]

1. The task force commander selects a feasible COA.
2. The operations officer selects the timing of battlefield activities, and gives detailed guidance to the engineer officer regarding the engineer tasks required to support the plan.
3. The engineer officer prepares an <u>engineer plan</u> to accomplish the necessary tasks prior to the battle.

Figure 3.1 illustrates a sample engineer plan superimposed on a portion (vicinity of engagement area New York) of the COA presented in Chapter II. Generally, engineer tasks are not co-located with battle positions (BP's). Instead, they tend

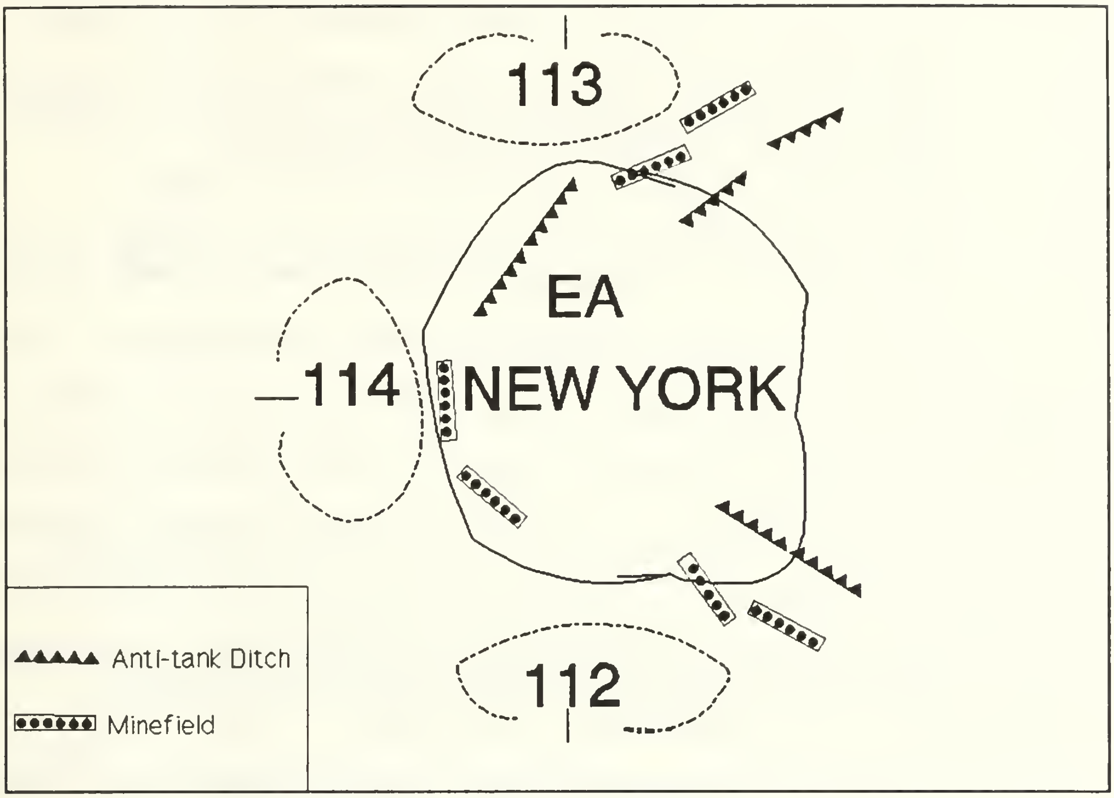

Figure 3.1 Sample Engineer Plan

to be near BP's. In the figure, two types of engineer tasks exist, survivability and countermobility. Survivability tasks include the preparation of positions to enhance the survivability of the units, and are, therefore, performed at the unit's battle position. Countermobility tasks include emplacing mine, wire, and anti-tank ditch obstacles and are not necessarily performed at the BP.

To prepare the engineer plan, the engineer officer, from his own expertise and from guidance given him by the operations officer, has the following information for preparing the battlefield.

1. The time required to complete the task at each location.
2. The priority associated with completion of the task at each location (our convention is to use a higher number for a higher priority position).
3. A 'combat value' associated with the task at each location.
4. The time by which the engineers must leave a location.
5. The type of engineer asset required at each location.

In general, the 'combat value' associated with the completion of the task at each location is a qualitative assessment of the relative worth of the different levels of completion of the task. In the simplest of terms, a partially prepared location is not as 'valuable' as a fully prepared one. This relationship between combat value and levels of completion is typically non-linear. In modeling this relationship, it is assumed that the time spent at a location directly corresponds to the level of completion, i.e., more time means more work is completed. Moreover, the combat value is assumed to vary in a piecewise linear fashion with time. A typical combat value function is given in Figure 3.2. In this figure, the combat value is zero if the time spent at a location is less than thirty minutes and the combat value remains constant after ninety minutes. In other words, the amount of time to complete the task is ninety minutes, and additional time spent on this task is simply wasted. Note also that the slopes of successive linear pieces are decreasing. This is to account for the diminishing return often encountered in modeling combat tasks. To illustrate, consider the task of digging a survivability position for a

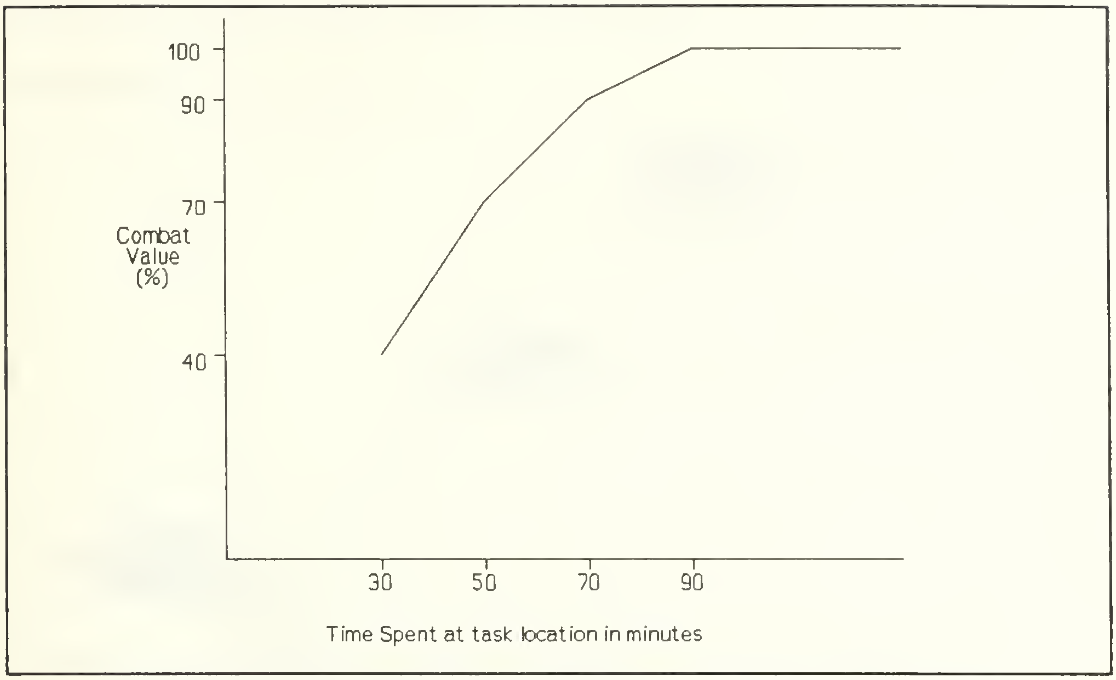

Figure 3.2 Piecewise Linear Combat Function

tank. There are three levels of constructing tank positions: hull defilade, turret defilade, and turret defilade with a hide position. [Ref 7:p. 4-9]

In a combat environment, tank positions at hull defilade greatly enhance tank survivability, and it is better to prepare all of the positions to one level before improving a single position to the next level. Thus, the additional time spent to obtain the turret defilade position is worth more than the time to prepare the hide position. Hence, the general form of the combat value function is similar to the one in Figure 3.4. If one of the engineer tasks is to 'dig in' a company of fourteen tanks, then the aggregated combat

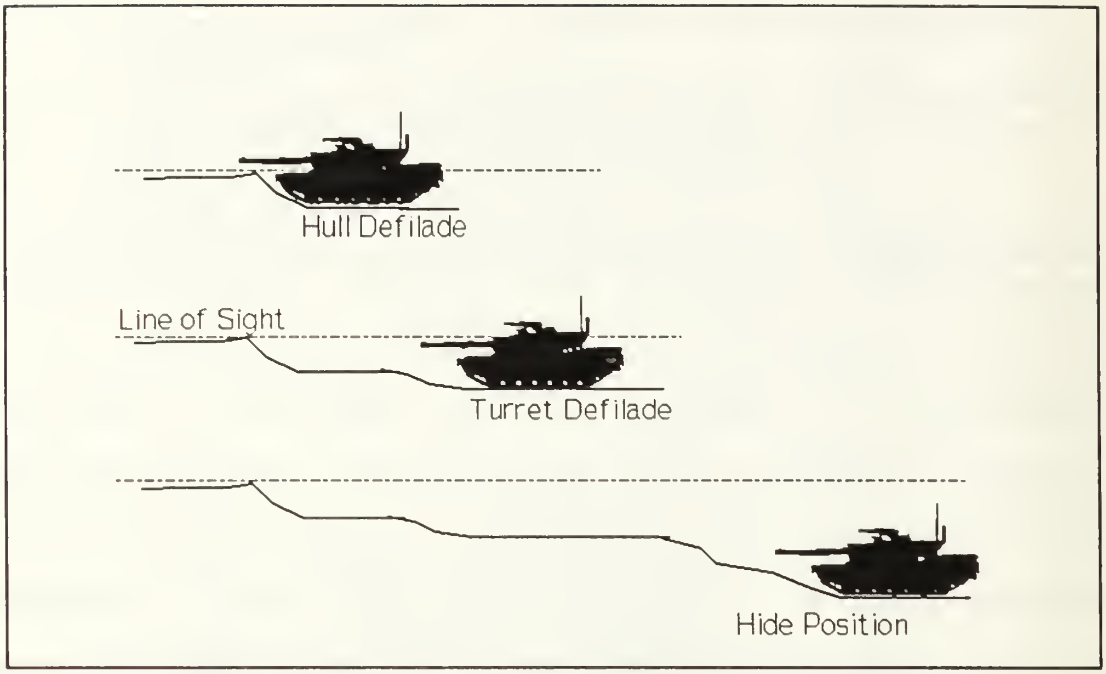

Figure 3.3 Position Construction Levels

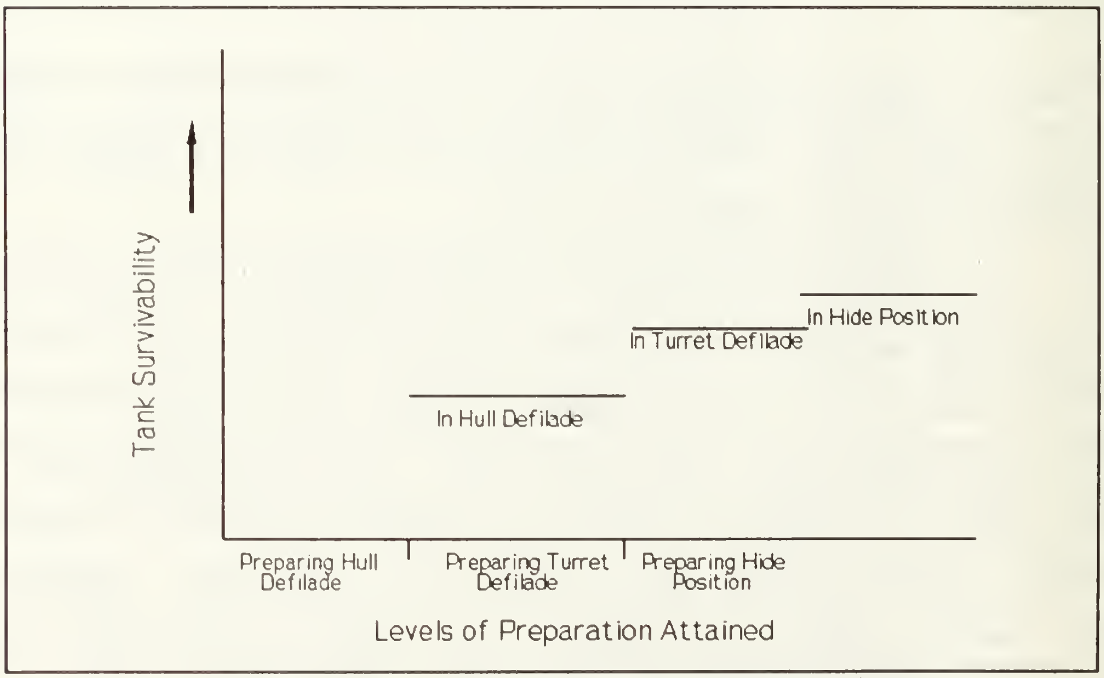

Figure 3.4 Single Tank Survivability

value function and its corresponding piecewise linear approximation are given in Figure 3.5.

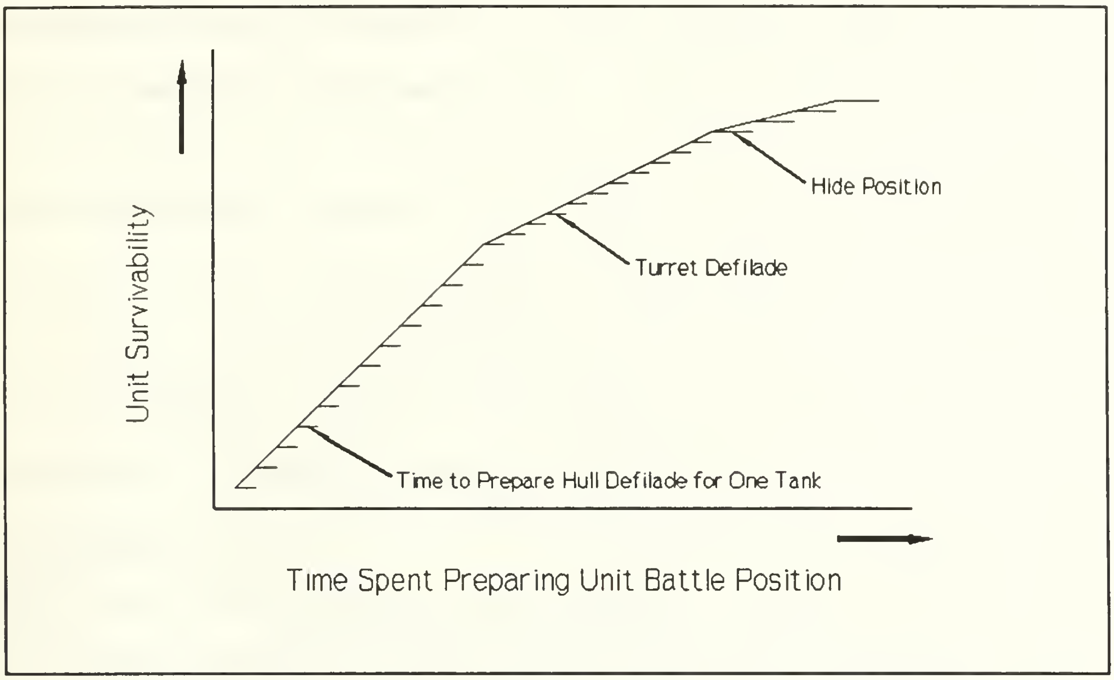

Figure 3.5 Aggregated Combat Value Function

With the available information, the engineer officer must prepare a schedule detailing where, when, and for how long each asset must perform the required tasks. In the next section, the underlying network structure of the model is described.

A. NETWORK STRUCTURE

In the formulation presented below, the movements of assets to the task locations are represented as networks. To illustrate, assume that there are three locations, numbered 1

to 3, and two assets, A and B. Furthermore, assume that A and B can perform the tasks at all three locations, and therefore, can visit any of the three. Figure 3.6 illustrates the networks which represent all possible movements of both assets.

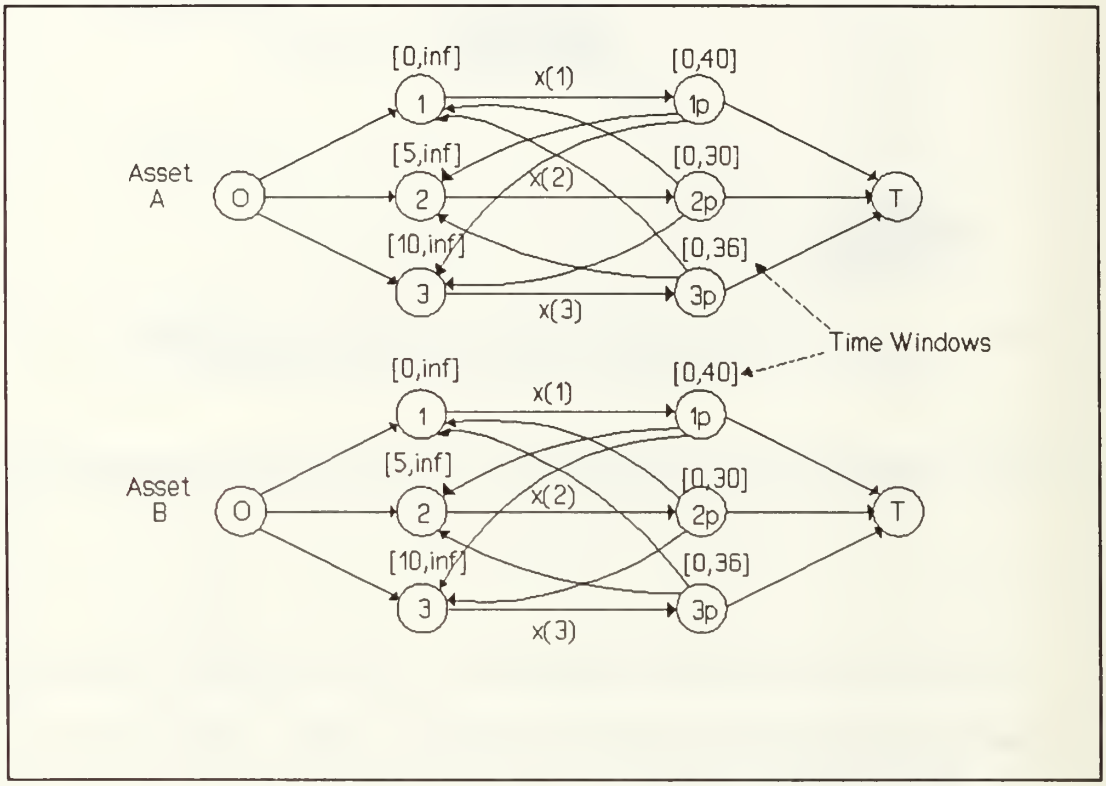

Figure 3.6 Engineer Model Network Structure

In the above networks, nodes O and T represent the origin and destination for all assets, i.e., they all start and end at the same place. Each location is represented with two nodes: the arrival node, denoted by i, and the departure node, denoted by ip. Arcs connecting nodes i to nodes ip, i.e., arcs (1,1p), (2,2p), and (3,3p) have variable length

which are denoted by x(i), and represent the time an asset spends performing the task at location i. Other arcs in the network represent the movement of assets and have associated travel times which are constant. It should be noted that the two networks in the figure have the same structure. In fact, they represent exactly the same movements and locations. However, each is for a different asset and each node, except for O and T, can be visited by at most one asset. In addition, there is a time constraint imposed on each arrival and departure node to insure the synchronization with other activities in the other battlefield operating systems (BOS). These time constraints are depicted as (time) intervals above the nodes in Figure 3.6. Thus, a feasible schedule for an asset corresponds to an acyclic path from O to T that arrives at the nodes within the specified time intervals.

B. AN INTEGER PROGRAMMING FORMULATION

Indices:

i,j	arrival nodes for each location, where i,j=1,2,3,...,I,O,T (O = origin node and T = destination node)
ip,jp	departure nodes for each location, where i,j=1,2,3,...,I
k	engineer assets, k = 1,2,3,...,K
q	linear functions in the piecewise linear combat value function, q = 1,2,3,...,Q

Given and Derived Data:

e_i earliest arrival to location i (or arrival node i departure)

d_{ip} latest departure time from location i (or departure node ip)

D_i maximum time to spend at location i

P_i priority for task at location i

t_{ij} travel time from location i (node ip) to location j

C_{ik} 1 if asset k is compatible with the task at location i; 0 otherwise

r_k factor to convert the work speed for asset k into standard speed

F latest arrival time at destination (i.e. node T)

a_{iq} intercept for the q^{th} linear piece of the combat value function for the task at location i

b_{iq} slope for the q^{th} linear piece at location i

m_{iq} maximum time to spend in the q^{th} linear piece at location i

Φ large constant

Variables:

$x^k{}_i$ total amount of time for asset k to spend at location i

$s^k{}_i$ arrival time for asset k at location i or ip

w_{iq} amount of time to spend in the q^{th} linear piece at location i

$z^k{}_{ij}$ 1 if asset travels the arc from node i (or ip) to node j (or jp); 0 otherwise

Formulation:

(Note that only compatible combinations of assets and locations are permitted in this formulation)

$$\textit{Maximize}: \sum_{i=1}^{I} P_i \sum_{q=1}^{Q} (b_{i,q} w_{i,q} + a_{i,q})$$

Subject to:

$$\sum_{i=1}^{I} z^k{}_{O,i} = 1 \qquad \forall\ k,\ i \neq O, T \qquad (1)$$

$$\sum_{i=1}^{I} z^k{}_{ip,T} = 1 \qquad \forall\ k \qquad (2)$$

$$z^k{}_{O,i} + \sum_{j \neq i}^{J} z^k{}_{jp,i} = z^k{}_{i,ip} \qquad \forall\ k,\ i \neq O, T \qquad (3)$$

$$z^k{}_{i,ip} = \sum_{j \neq i}^{J} z^k{}_{ip,j} \qquad \forall\ k,\ i \neq O, T \qquad (4)$$

$$s^k{}_O + t_{O,i} \leq s^k{}_i + \Phi(1 - z^k{}_{O,i}) \qquad \forall\ k,\ i \qquad (5)$$

$$s^k{}_i + x^k{}_i \leq s^k{}_{ip} + \Phi(1 - z^k{}_{i,ip}) \qquad \forall\ k,\ i \qquad (6)$$

$$s^k{}_{ip} + t_{i,j} \leq s^k{}_j + \Phi(1 - z^k{}_{ip,j}) \qquad \forall\ k,\ i,\ j \neq O \qquad (7)$$

$$\sum_{q=1}^{Q} w_{i,q} = \sum_{k=1}^{K} r_k x^k{}_i \qquad \forall\ i \neq O, T \qquad (8)$$

$$w_{i,1} = a_{i,1} \sum_{k=1}^{K} z^k_{i,ip} \qquad \forall\ i \neq O, T \qquad (9)$$

$$w_{i,q} \leq (m_{i,q} - m_{i,q-1}) \sum_{k=1}^{K} z^k_{i,ip} \qquad \forall\ k, i \qquad (10)$$

$$\sum_{k=1}^{K} z^k_{i,ip} \leq 1 \qquad \forall\ i \qquad (11)$$

$$x^k_i \leq D_i \sum_{k=1}^{K} z^k_{i,ip} \qquad \forall\ i \qquad (12)$$

$$s^k_i \geq e_i \qquad \forall\ k, i \qquad (13)$$

$$s^k_{ip} \leq d_{ip} \qquad \forall\ k, i \qquad (14)$$

$$s^k_T \leq F \qquad \forall\ k \qquad (15)$$

$$s^k_i \geq 0,\ x^k_i \geq 0,\ w_{iq} \geq 0,\ z^k_{i,ip} = 0, 1$$

The objective is to maximize the weighted combat values of the tasks, here represented by the given priority multiplied by the amount of work accomplished within each interval for the task. Constraints (1) through (4) represent the balance of flow constraints with constraints (1) and (2) being the

flow out of the origin and into the destination, respectively. An equality is used in these two constraints to ensure that each asset is utilized. Constraint (3) ensures that the flow balance is maintained at the arrival nodes, while constraint (4) ensures the flow balance at the departure nodes.

Constraints (5) through (7) calculate arrival times at the origin, arrival, departure and destination (T) nodes. In constraint (5), if $z^k_{0,i}$ equals 1, i.e., asset k travels from node O to node i, then the start time at node O plus the travel time to position i, must be less than or equal to the arrival time at (arrival) node i. If $z^k_{o,i}$ equals 0, then this constraint is not binding because of the large constant, Φ. Similarly, constraints (6) and (7) calculate arrival times for the nodes incident to arcs of the form (i,ip), and (ip,j) for $i \neq j$, respectively.

Constraints (8) through (10) comprise those necessary for the use of the piecewise linear function. Constraint (8) simply requires that the sum of the work done in each interval equals the total amount of work done at the location. Constraint (9) ensures that at least the minimum amount of work, i.e., $a_{i,1}$, is done at location i, while constraint (10) ensures that the maximum amount of work done in each interval is not exceeded.

Constraint (11) ensures that each position is visited by at most one asset. If $z^k_{i,ip}$ equals one for some asset k, then

only this asset is permitted to do work at this position, while $z^k_{i,ip}$ for all other k becomes zero.

Constraint (12) ensures that an asset spends no longer than the required time to complete a task at a position it visits. Constraints (13) and (14) simply ensure that the each asset arrives at (node i) and departs from (node ip) its task location within the specified time windows. Constraint (15) ensures that all assets arrive at the final destination by the required time.

IV. MODEL IMPLEMENTATION

The Maneuver and Engineer models were both implemented in the General Algebraic Modeling System (GAMS). [Ref 8] The solver for the Maneuver model is the linear programming software developed by Brooke, et al. [Ref 8:p. xiii] For the Engineer model, an integer programming solver, Zero/One Optimization Methods (ZOOM), was used. [Ref 9] Both models were solved on the AMDAHL 5990 computer at the Naval Postgraduate School.

The two models were validated using a small data set with known solutions. To demonstrate the model effectiveness on a realistic scenario, the framework for a tactical scenario was obtained from Major William Odom, a current operations officer for an infantry battalion at Fort Ord, California. The scenario requires the task force to defend against an attack of a Soviet style motorized rifle regiment. From the receipt of the order from the parent brigade, the task force has seventy two hours available for planning and preparing the battlefield prior to the enemy attack. The task force is organized with two tank companies, one mech-heavy team (two platoons of mechanized infantry and one platoon of tanks), and one tank-heavy team (two platoons of tanks and one platoon of mechanized infantry).

A. MANEUVER MODEL

The main input data for the Maneuver model is the critical activities and their associated critical times. Both of these will impact the course of action to be considered by the task force. For the output shown in Figure 4.1, a course of action similar to the one in Chapter 2 was used.

In this scenario, one unit remains stationary throughout the battle, one unit repositions to a new BP once, and the remaining two units reposition twice. The COA allows the task force to fight the enemy throughout the depth of the battlefield. The times for the critical activities were chosen based on a plausible course of events. When all parameters for the scenario are specified, the resulting model contains 23 activities and 5 critical events. The corresponding linear program contains 33 variables and 33 constraints. The problem was solved in .05 CPU seconds. Figure 4.1 displays a portion of the output from the Maneuver model.

In Figure 4.1, the column labeled 'earliest' is the earliest time at which the specified activity can begin without being asynchronous with other activities. Likewise, 'latest' is the latest time at which the activity can begin. The column labeled 'due' represents the critical time by which the activity must be completed. To illustrate, the first line in the table states that activity B-DEP-AA (B Company departs

the assembly area) can commence during the time interval between hour 0 (a null entry in the figure) and hour 7.85.

Given the output in Figure 4.1, the operations officer would select a time in the interval between the earliest and the latest time to be the time at which the activity would begin. In most cases, the midpoints of the intervals were chosen for Engineer model input. However, for intervals less than thirty minutes, the earliest time was chosen instead.

	EARLIEST	DUE	LATEST
B-DEP-AA			7.850
B-M-OC-101	0.500	12.000	9.500
A-DEP-AA	0.500		8.350
A-M-OC-103	1.000	12.000	10.000
B-OCC-121	74.750	75.500	75.300
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
A-DEP-113	74.500		75.050
A-MV-123	74.600		75.150
A-OCC-123	74.750	75.500	75.300

Figure 4.1
Maneuver Model Output

B. ENGINEER MODEL

The next step in the synchronization process is to take the times chosen above and use them as times at which the various engineer tasks must be completed. It is the responsibility of the engineer officer to ensure that all tasks are completed on time.

The above scenario contains thirty engineer tasks which include minefield emplacement, construction of wire obstacles, and digging vehicle positions and anti-tank ditches. Of the thirty total tasks, thirteen are vehicle positions and anti-tank ditches and require the available blade assets which include Armored Combat Earthmovers (ACE) and Combat Engineer Vehicles (CEV). The task force has six ACE's and two CEV's available. These are paired to form teams of two ACE's or CEV's, and each team is considered as one asset. [Ref 10:p. M/CM/S-22] The distance between task locations varies between 0 and 33 kilometers, and the time required to complete the positions ranges from 4 to 17 hours. The ACE teams and CEV team can perform any of the thirteen tasks requiring blade assets, and their movement and work rates are identical.

The remaining seventeen tasks consist of minefield emplacement and construction of wire obstacles and require either an engineer platoon or a Ground Emplaced Mine Scattering System (GEMSS) which is essentially an automatic mine dispenser. The task force has three engineer platoons and one GEMSS. The movement rates for these two types of assets were the same, but the work rate for the GEMSS was ten times that of one platoon. Because of special considerations for employing GEMSS, this asset could only be used for five minefields. The engineer platoons could be used for any of the seventeen tasks.

The resulting integer program contains 2335 constraints, 453 continuous variables, and 1784 binary variables. Due to the compatibility of assets and tasks, the integer program can be decomposed into two independent programs. One subprogram is for scheduling blade assets to prepare vehicle positions and anti-tank ditches, and the other is for scheduling the platoons and GEMSS to perform mine emplacement and wire obstacle construction. With the decomposition, the blade portion of the model has 1,036 constraints, 117 continuous variables, and 780 binary variables, while the remaining portion has 1300 constraints, 245 continuous variables, and 1004 binary variables. The output of these two models are in Figures 4.2 and 4.3, respectively.

	ARRTIME	EFFORT	DEPART	DUE	MAINT	PRIORITY	DESTTIME
1 .ACE1-ACE2	54.00	10.00	66.00	72.00	2.00	5.00	67.32
2 .ACE1-ACE2	27.97	11.00	42.97	72.00	4.00	9.00	67.32
3 .ACE1-ACE2	43.22	8.48	53.70	72.00	2.00	9.00	67.32
4 .CEV1-CEV2	37.21	17.00	60.21	72.00	6.00	8.00	61.43
5 .ACE5-ACE6	0.58	7.00	7.58	72.00		8.00	10.76
6 .ACE3-ACE4	13.89	10.00	25.89	72.00	2.00	9.00	27.16
7 .CEV1-CEV2	13.37	10.00	25.37	72.00	2.00	3.00	61.43
8 .CEV1-CEV2	25.83	9.00	36.83	73.50	2.00	9.00	61.43
9 .ACE1-ACE2	17.91	7.70	27.61	73.50	2.00	6.00	67.32
10.ACE3-ACE4	0.44	11.00	13.44	73.50	2.00	6.00	27.16
11.CEV1-CEV2	0.70	10.00	12.70	75.00	2.00	9.00	61.43
12.ACE1-ACE2		7.70	7.70	75.00		9.00	67.32
13.ACE1-ACE2	7.85	7.70	17.55	75.00	2.00	9.00	67.32

Figure 4.2 Engineer Model Output (Blade Assets)

	ARRTIME	EFFORT	DEPART	DUE	MAINT	PRIORITY	DESTTIME
1 .PLT1	18.07	4.00	22.07	72.00		2.00	68.32
2 .PLT1	0.76	7.00	7.76	72.00		5.00	68.32
3 .PLT1	7.90	7.00	17.90	72.00	3.00	7.00	68.32
4 .PLT3	8.52	10.00	21.52	72.00	3.00	7.00	22.48
5 .PLT1	22.29	8.40	33.69	72.00	3.00	8.00	68.32
6 .PLT2	30.72	4.00	34.72	73.50		2.00	35.69
7 .GEMSS	0.39	0.80	1.19	73.50		2.00	6.13
8 .PLT1	41.06	2.80	46.86	73.50	3.00	2.00	68.32
9 .GEMSS	1.29	0.80	2.09	73.50		3.00	6.13
10.PLT3	0.26	8.00	8.26	73.50		8.00	22.48
11.PLT1	33.89	7.00	40.89	73.50		6.00	68.32
12.PLT2	17.37	10.00	30.37	73.50	3.00	5.00	35.69
13.PLT1	60.50	7.00	67.50	73.50		5.00	68.32
14.PLT1	47.28	9.95	60.23	75.00	3.00	8.00	68.32
15.PLT2		14.00	17.00	75.00	3.00	8.00	35.69
16.GEMSS	2.66	1.40	4.06	75.00		8.00	6.13
17.GEMSS	4.13	1.40	5.53	75.00		8.00	6.13

Figure 4.3 Engineer Model Output (Platoons/GEMSS)

The row labels indicate the task-asset assignment. For example, the blade unit, ACE1-ACE2, is assigned to task in the first row. The first three columns indicate the arrival time of the asset at the location, the amount of time spent working, and the departure time, respectively. The 'due' column represents the time by which the task must have been completed. The fifth column, maintenance, shows the amount of time allocated for maintenance or rest while at the position. The last two columns give the priority of the tasks and the time at which the assets arrive at the final destination.

Allowing for maintenance and rest in the model can be accomplished by adjusting the due time of the task using equation 4.1.

In equation 4.1, s^k_i denotes the time by which asset k must complete task i, and d_i is the time (in hours) by which the

$$s^{k}_{i} = d_i - INTEGER\left(\frac{d_i}{24} * MP_k\right) * MD_k \tag{4.1}$$

task must be completed (or due time). MP_k and MD_k are the number and length of maintenance (or rest) periods required per day by asset k, respectively. The expression inside the brackets gives the number of maintenance periods that should be completed by the time d_i. Taking the integer part of this expression prohibits fractional maintenance periods and multiplying this value by MD_k gives the time allocated to maintenance up to the time d_i. Subtracting this quantity from the given due time, in effect, establishes a new upper bound on the arrival time at that location. These new arrival times will force assets to arrive at the position with sufficient time to complete the tasks at the desired level, to perform maintenance and to provide rest.

As stated in Chapter 3, the objective of the Engineer model is to maximize the weighted combat values of the tasks completed within the specified amount of time. When the specified time is excessive, the model lets the assets 'loiter' around a task location and arrive at the final destination exactly at the end of the specified time. This is unacceptable in practice. The assets should arrive at the final destination as soon as they complete all the tasks assigned to them. To remedy this problem, another integer programming (IP) problem is constructed to ensure that all

assets arrive at the destination as soon as they are finished with their tasks. This IP problem essentially eliminates the loiter time resulting from the Engineer model. Based computational experience, the time required to solve both the Engineer and new IP models is less than the time to solve the Engineer model in which the assets are penalized for unnecessarily late arrivals.

Since the Engineer model is an integer program, a class of problems which is generally time consuming to solve optimally, a small experiment was conducted to observe the computational time for several problem sizes. Tables 4.1 and 4.2 display the CPU times required for ZOOM to produce solutions within ten percent and one percent of the best linear programming lower bound. From these tables, the difference in CPU times at two different optimality levels is slight when the number of tasks is between 7 and 12. However, ZOOM required an unreasonable amount of time to achieve a one percent optimality level for the problem with 13 tasks. Moreover, ZOOM may not be appropriate for problems with 3 assets and more than 11 tasks, if the model is implemented on a microcomputer. Although this small experiment may not be conclusive, it points out that the appropriate degree of optimality would depend on the type of computer and solver used to solve the model. Two alternate integer solvers which deserve consideration are recommended for future investigation.

TABLE 4.1
WITHIN 10 PERCENT OF OPTIMALITY

Number of Tasks	Number of Assets	Objective Value	Percent from Optimality	CPU Time (in sec)
7	3	193.92	9.77	16.10
8	3	221.55	9.43	30.51
9	3	264.52	1.31	26.02
10	3	289.15	0.41	48.97
11	3	319.58	1.31	61.23
1[illegible]	3	329.72	8.14	117.46
13	3	370.95	5.86	379.82

TABLE 4.2
WITHIN 1 PERCENT OF OPTIMALITY

Number of Tasks	Number of Assets	Objective Value	Percent from Optimality	CPU Time (in sec)
7	3	214.77	0.57	19.04
8	3	243.62	0.41	47.04
9	3	268.02	0.00	26.08
10	3	291.23	0.07	49.81
11	3	323.82	0.00	83.78
1[illegible]	3	356.86	0.57	985.74
13	3	N/A	N/A	>20,000

C. ENGINEER MODEL FLEXIBILITY

By varying the data inputs to both the Maneuver and Engineer models, one can observe the relationship between various data input and model output. To illustrate, the number of assets available for 17 mine emplacements and wire obstacle constructions were varied. The results are displayed Table 4.3. Note that as expected, the average level of completion of the tasks attempted increases with the available number of assets. This table provides practical information to the engineer officer if he associates probability to the table. For example, based on some reliability analysis, one can compute the probability, p_k, of having k, where k=1,2,3, assets in working condition. Then, the following relationship exists:

$$\text{Prob[average level of completion} \geq 42.2\%] = p_1+p_2+p_3$$
$$\text{Prob[average level of completion} \geq 77.5\%] = p_2+p_3$$
$$\text{Prob[average level of completion} \geq 97.2\%] = p_3$$

If $p_2 + p_3$ is .95, then the engineer officer is '95% sure' that the tasks will be completed at a level of 77.5% or more.

TABLE 4.3
EFFECT OF FEWER AVAILABLE ASSETS

Number of Assets	Number of Tasks Planned	Average Completion of Tasks Attempted (%)			Percent from Optimal
		Min	Avg	Max	
1	17	40	42.2	70	8.73
2	17	40	77.5	100	4.18
3	17	90	97.2	100	0.52

V. CONCLUSIONS

This thesis discusses the formulation of two optimization models to aid a task force in Army battlefield synchronization planning. The first model is called the Maneuver model. It integrates the first two battlefield operating systems (BOS), Intelligence and Maneuver, in order to determine feasibility of a given course of action (COA). If the COA is feasible, the Maneuver model then calculates for each activity the earliest and latest times at which the activity can begin. The second model, the Engineer model, synchronizes the activities of the third battlefield operating system, Mobility/Countermobility/ Survivability, with the first two.

The use of the two models automates and optimizes an important combat planning process which is currently conducted manually. The models also accommodate the changes that frequently occur on a fluid battlefield by providing rapid responses to different situations. 'What if' analysis is possible by varying the input of both models. Input to the Maneuver model can be varied to determine the effects of changes in potential enemy attack times. Varying Engineer model input can determine the effects of losing or gaining engineer assets. Finally, the models can be integrated with models of similar structure from the other BOS.

The results of this thesis also point out several areas for further investigation. They are as follows:

1) The models in this thesis are deterministic and accommodate one given situation at a time. However, techniques such as stochastic programming would allow models to consider several courses of action, each having a probability of occurrence.

2) ZOOM, the integer program solver used in this thesis, proved to be time consuming in many instances. Other solvers, such as the X-System [Ref 11] and the XA Professional Linear Programming System produced by Sunset Software should be investigated.

3) In the long run, a complete army battlefield synchronization system should be developed. This system should include models for all seven battlefield operating systems (BOS) mentioned in Chapter 1 and have a graphical user interface to facilitate data input as well as user acceptance. The thesis addresses the first three BOS. However, the models for the remaining four BOS can be developed in a similar manner. As for the interface, it should use appropriate military graphic symbols and generate the necessary data for the models. The latter would require the system to maintain a database of, e.g., doctrinal times for activities such as movements and preparing positions.

In summary, synchronization of battlefield activities is critical for the task force when conducting combat operations.

The level of detail required to effectively utilize available assets places extreme demands on the staff officer preparing plans for an operation. A tool which is automated and based on optimization models can aid not only the operations officer and others in their decision making, but it also enhances the probability of successfully completing the required mission on the battlefield.

APPENDIX A: GAMS INPUT FOR SAMPLE MANEUVER PROBLEM

1. DATA

```
*-----------------------------------------------------------------
*                        INDICES
*-----------------------------------------------------------------
SETS

   I    Task Force Movements
        /B-DEP-AA,B-M-OC-101,A-DEP-AA,A-M-OC-103
         C-DEP-AA,C-M-OC-102,D-DEP-AA,D-M-OC-104
         B-DEP-101,B-MV-111,B-OCC-111,A-DEP-103
         A-MV-113,A-OCC-113,C-DEP-102,C-MV-122
         C-OCC-122,B-DEP-111,B-MV-121,B-OCC-121
         A-DEP-113,A-MV-123,A-OCC-123/

   K    Critical Activities

        /ISS-OPORD,M-B-FORD,M-B-CHEVY
         M-B-OLDS,M-B-BUICK/;

ALIAS(I,J);

*-----------------------------------------------------------------
*                        GIVEN DATA
*-----------------------------------------------------------------
PARAMETERS

   T(I) Duration time for task force movement i

        /B-DEP-AA    .5, B-M-OC-101     2.5
         A-DEP-AA    .5, A-M-OC-103     2.0
         C-DEP-AA    .5, C-M-OC-102     1.9
         D-DEP-AA    .5, D-M-OC-104     2.15
         B-DEP-101   .1, B-MV-111        .2
         B-OCC-111   .1, A-DEP-103       .2
         A-MV-113    .15,A-OCC-113       .1
         C-DEP-102   .2, C-MV-122        .35
         C-OCC-122   .2, B-DEP-111       .1
         B-MV-121    .15,B-OCC-121       .2
         A-DEP-113   .1, A-MV-123        .15
         A-OCC-123   .2/
```

```
S(K) Start time for critical activities

     /ISS-OPORD     12.0
      M-B-FORD      73.0
      M-B-CHEVY     74.00
      M-B-OLDS      74.50
      M-B-BUICK     75.50/

P(I,J)  Task force activity i precedes critical activity j

     /B-DEP-AA.B-M-OC-101    1,B-DEP-AA.A-DEP-AA    1
      A-DEP-AA.A-M-OC-103    1,A-DEP-AA.C-DEP-AA    1
      C-DEP-AA.C-M-OC-102    1,C-DEP-AA.D-DEP-AA    1
      D-DEP-AA.D-M-OC-104    1,B-DEP-101.B-MV-111   1
      B-MV-111.B-OCC-111     1,B-DEP-101.C-DEP-102  1
      A-DEP-103.A-MV-113     1,A-MV-113.A-OCC-113   1
      A-DEP-103.C-DEP-102    1,C-DEP-102.C-MV-122   1
      C-MV-122.C-OCC-122     1,B-DEP-111.B-MV-121   1
      B-MV-121.B-OCC-121     1,A-DEP-113.A-MV-123   1
      A-MV-123.A-OCC-123     1/

Q(I,K)  Task force activity i precedes critical activity k

      /A-M-OC-103.ISS-OPORD  1
       B-M-OC-101.ISS-OPORD  1
       C-M-OC-102.ISS-OPORD  1
       D-M-OC-104.ISS-OPORD  1
       B-OCC-111.M-B-CHEVY   1
       A-OCC-113.M-B-CHEVY   1
       A-OCC-123.M-B-BUICK   1
       B-OCC-121.M-B-BUICK   1
       C-OCC-122.M-B-BUICK   1/

R(K,I)  Critical activity k precedes task force activity i
      /M-B-FORD.B-DEP-101    1
       M-B-FORD.A-DEP-103    1
       M-B-OLDS.B-DEP-111    1
       M-B-OLDS.A-DEP-113    1/;
```

2. GAMS FORMULATION

```
$TITLE                    Maneuver Model

*------------------------------------------------------------
*                    GAMS OPTIONS SETTINGS
*------------------------------------------------------------
$OFFUPPER OFFSYMXREF OFFSYMLIST

 OPTIONS LIMCOL = 0, LIMROW = 0, SOLPRINT = OFF;
 OPTIONS RESLIM = 100, ITERLIM = 10000;
```

```
*------------------------------------------------------------------
*                         GIVEN DATA
*------------------------------------------------------------------
$INCLUDE MANSCEN DATA A

*------------------------------------------------------------------
*                      DECISION VARIABLES
*------------------------------------------------------------------
POSITIVE VARIABLES

  X(I)       Start time of Blue activity i
  Z(I)       Additional time needed to complete activity i;

       X.UP(I) = SUM(K, S(K));
       Z.UP(I) = SUM(K, S(K));

 VARIABLE

  START      Sum of start times;

*------------------------------------------------------------------
*                         FORMULATION
*------------------------------------------------------------------
 EQUATIONS

  TIMES      Sum of start times
  TF(I,J)    Precedence of task force activities
  TFPC(I,K)  Task force activity precedes critical activity
  CPTF(K,I)  Critical activity precedes task force activity;

   TIMES..          START =E= SUM(I, X(I));

   TF(I,J) $ (P(I,J) EQ 1)..
                  X(I) + T(I) - X(J) =L= 0;

   TFPC(I,K) $ (Q(I,K) EQ 1)..
                  X(I) + T(I) - Z(I) - S(K) =L= 0;

   CPTF(K,I) $ (R(K,I) EQ 1)..
                  X(I) - S(K) =G= 0;


 MODEL MANEUVER /ALL/;

 SOLVE MANEUVER USING LP MINIMIZING START;

 PARAMETER SOLN(*,*);
 SOLN(I,'EARLIEST') = X.L(I);
 SOLN(I,'DUE') =  SUM(K $ (Q(I,K) EQ 1), S(K));
 PARAMETER INFEAS(I);
 INFEAS(I) = Z.L(I);
```

```
Z.FX(I) = 0;

SOLVE MANEUVER USING LP MAXIMIZING START;
SOLN(I,'LATEST') = X.L(I);


DISPLAY SOLN, INFEAS;
```

APPENDIX B: GAMS INPUT FOR SAMPLE ENGINEER PROBLEM

1. MINEFIELD AND WIRE ASSET DATA

```
*-----------------------------------------------------------------
*                          INDICES
*-----------------------------------------------------------------
SETS

   V       All nodes in the model network
           /1*17,O,T,1P*17P/

   I(V)    'Origin, arrival, and destination nodes'
           /1*17,O,T/

   IP(V)   Departure nodes
           /1P*17p/

   ITV     Intervals for piecewise linear functions
           /1*4/

   K       Assets
           /PLT1, PLT2, PLT3, GEMSS/;

ALIAS(I,J);

*-----------------------------------------------------------------
*                     GIVEN DATA
*-----------------------------------------------------------------
PARAMETERS

    T(IP)  Time by which activity i must be completed
           / 1P   72.0,  2P   72.0,  3P   72.0,  4P   72.0,
             5P   72.0,  6P   73.5,  7P   73.5,  8P   73.5,
             9P   73.5, 10P   73.5, 11P   73.5, 12P   73.5,
            13P   73.5, 14P   75.0, 15P   75.0, 16P   75.0,
            17P   75.0/


    D(I)   Time required to complete task i in hours
           / 1      8,  2     10,  3     10,  4     10,  5     12,
             6      4,  7      8,  8      4,  9      8, 10      8,
            11     10, 12     10, 13     10, 14     10, 15     14,
            16     14, 17     14 /
```

```
  P(I)    Priority given by commander to task i
          / 1     2,  2     5,  3     7,  4     7,  5     8,
            6     2,  7     2,  8     2,  9     3, 10     8,
           11     6, 12     5, 13     5, 14     8, 15     8,
           16     8, 17     8/

  MNTPRD(K) Number of maintenance periods required per day
         /PLT1          2
          PLT2          2
          PLT3          2
          GEMSS         2/


 MNTDUR(K) Duration of maintenance period
         /PLT1          3
          PLT2          3
          PLT3          3
          GEMSS         1/


  FACTOR(K) Ratio of work speed of asset k to standard asset
         /PLT1          1
          PLT2          1
          PLT3          1
          GEMSS        10/

TABLE

 LEVEL(I,ITV)  Percent of combat value attained per interval

             1     2     3     4
    1       .4    .7    .9     1
    2       .4    .7    .9     1
    3       .4    .7    .9     1
    4       .4    .7    .9     1
    5       .4    .7    .9     1
    6       .4    .7    .9     1
    7       .4    .7    .9     1
    8       .4    .7    .9     1
    9       .4    .7    .9     1
   10       .4    .7    .9     1
   11       .4    .7    .9     1
   12       .4    .7    .9     1
   13       .4    .7    .9     1
   14       .4    .7    .9     1
   15       .4    .7    .9     1
   16       .4    .7    .9     1
   17       .4    .7    .9     1;
```

```
TABLE
   WORK(I,ITV) Maximum work time per interval in hours

                1      2      3      4
      1        2.4    4.0    5.6    8.0
      2        3.0    5.0    7.0   10.0
      3        3.0    5.0    7.0   10.0
      4        3.0    5.0    7.0   10.0
      5        3.6    6.0    8.4   12.0
      6        1.2    2.0    2.8    4.0
      7        2.4    4.0    5.6    8.0
      8        1.2    2.0    2.8    4.0
      9        2.4    4.0    5.6    8.0
     10        2.4    4.0    5.6    8.0
     11        3.0    5.0    7.0   10.0
     12        3.0    5.0    7.0   10.0
     13        3.0    5.0    7.0   10.0
     14        3.0    5.0    7.0   10.0
     15        4.2    7.0    9.8   14.0
     16        4.2    7.0    9.8   14.0
     17        4.2    7.0    9.8   14.0;

TABLE
   DIST(I,J)  Distance from one position to another in Km

         1      2      3      4      5
  1      0      .8    5.0    5.8    6.6
  2             0     4.4    5.3    5.8
  3                   0      1.2    1.5
  4                          0      1.2
  5                                 0
  O    16.3   15.8   14.3   14.3   13.3

   +     6      7      8      9      10     11
  1    11.5   11.5   10.4   10.4    9.8   10.0
  2    11.0   11.0    9.9    9.9    9.3    9.5
  3     9.2    9.2    8.1    8.1    7.5    7.7
  4     9.5    9.5    8.4    8.4    7.8    8.0
  5     7.5    7.5    6.4    6.4    5.8    6.0
  6     0      0      1.0    1.0    2.0    3.5
  7            0      1.0    1.0    2.0    3.5
  8                   0      0      1.0    2.5
  9                          0      1.0    2.5
 10                                 0      1.5
 11                                        0
  O     9.7    9.7   10.7   10.7    7.7    6.2
```

+	12	13	14	15	16	17	T
1	6.3	7.6	14.0	16.3	16.6	18.6	30.8
2	5.6	6.8	13.5	15.8	16.1	18.1	30.3
3	6.0	6.3	12.0	14.3	14.1	16.1	28.8
4	7.2	7.5	12.0	14.3	14.1	16.1	28.8
5	6.0	6.2	11.0	13.3	13.1	15.1	27.8
6	7.0	6.5	7.5	9.7	10.5	12.5	24.2
7	7.0	6.5	7.5	9.7	10.5	12.5	24.2
8	6.0	5.5	8.5	10.7	11.5	13.5	25.2
9	6.0	5.5	8.5	10.7	11.5	13.5	25.2
10	5.3	4.7	5.5	7.7	8.5	10.5	22.2
11	5.0	4.2	4.0	6.2	7.0	9.0	20.7
12	0	1.3	9.0	11.2	12.0	14.0	25.7
13		0	8.0	10.2	11.0	13.0	24.7
14			0	2.2	3.0	5.0	16.7
15				0	1.5	3.5	14.5
16					0	2.0	16.0
17						0	18.0
O	11.2	10.2	2.2	0	1.5	3.0	0 ;

TABLE

RATE(I,J) Rate of movement from i to j in Kmh

	1	2	3	4	5
1	0	30	30	30	30
2		0	30	30	30
3			0	30	30
4				0	30
5					0
O	30	30	30	30	30

+	6	7	8	9	10	11
1	20	20	20	20	30	30
2	20	20	20	20	30	30
3	20	20	20	20	30	30
4	20	20	20	20	30	30
5	20	20	20	20	30	30
6	0	10	10	10	15	15
7		0	10	10	15	15
8			0	10	15	15
9				0	15	20
10					0	30
11						0
O	25	25	25	25	30	30

```
  +    12    13    14    15    16    17     T
  1    30    30    30    30    30    30    30
  2    30    30    30    30    30    30    30
  3    30    30    30    30    30    30    30
  4    30    30    30    30    30    30    30
  5    30    30    30    30    30    30    30
  6    20    20    20    20    20    20    25
  7    20    20    20    20    20    20    25
  8    20    20    20    20    20    20    25
  9    20    20    20    20    20    20    25
 10    30    30    30    30    30    30    30
 11    30    30    30    30    30    30    30
 12     0    30    30    30    30    30    30
 13           0    30    30    30    30    30
 14                 0    25    30    30    30
 15                       0    30    30    30
 16                             0    30    30
 17                                   0    30
  O    30    30    30    30    30    30     0;
TABLE

   COMPAT(K,I)   Asset k can be used at position i

         1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
 PLT1    1  1  1  1  1  1  1  1  1   1   1   1   1   1   1
 PLT2    1  1  1  1  1  1  1  1  1   1   1   1   1   1   1
 PLT3    1  1  1  1  1  1  1  1  1   1   1   1   1   1   1
 GEMSS   1  0  0  0  0  0  1  0  1   0   0   0   0   0   0

 +          16  17
 PLT1        1   1
 PLT2        1   1
 PLT3        1   1
 GEMSS       1   1;


 SCALAR   FINISH   Time to arrive at destination      /78/;


2.  BLADE ASSET DATA
*-----------------------------------------------------------
*                          INDICES
*-----------------------------------------------------------
SETS

   V       All nodes in the model network
           /1*13,O,T,1P*13P/

   I(V)    'Origin, arrival, and destination nodes'
           /1*13,O,T/
```

```
   IP(V)   Departure nodes
           /1P*13p/

   ITV     Intervals for piecewise linear functions
           /1*4/

   K       Assets
           /ACE1-ACE2, ACE3-ACE4, ACE5-ACE6, CEV1-CEV2/;

ALIAS(I,J);
*-------------------------------------------------------------
*                    GIVEN DATA
*-------------------------------------------------------------
PARAMETERS

    T(IP)  Time by which activity i must be completed
          / 1P  72.0,2P   72.0,3P   72.0,4P   72.0,5P   72.0,
            6P  72.0,7P   72.0,8P   73.5,9P   73.5,10P  73.5,
            11P 75.0,12P  75.0,13P  75.0/

    D(I)   Time required to complete task i in hours

           / 1    10,2    11,3    11,4    17,5     7,
             6    10,7    10,8     9,9    11,10   11,
            11    10,12   11,13   11/

    P(I)   Priority given by commander to task i

           / 1     5,2     9,3     9,4     8,5     8
             6     9,7     3,8     9,9     6,10    6,
            11     9,12    9,13    9/

    MNTPRD(K) Number of maintenance periods required per day
          /ACE1-ACE2     3
           ACE3-ACE4     3
           ACE5-ACE6     3
           CEV1-CEV2     3/

    MNTDUR(K) Duration of maintenance period
          /ACE1-ACE2     2
           ACE3-ACE4     2
           ACE5-ACE6     2
           CEV1-CEV2     2/

    FACTOR(K) Ratio of work speed of asset k to standard asset
          /ACE1-ACE2     1
           ACE3-ACE4     1
           ACE5-ACE6     1
           CEV1-CEV2     1/
```

TABLE

LEVEL(I,ITV) Percent of combat value attained per interval

	1	2	3	4
1	.4	.7	.9	1
2	.4	.8	.95	1
3	.4	.8	.95	1
4	.4	.7	.9	1
5	.4	.7	.9	1
6	.4	.8	.95	1
7	.4	.7	.9	1
8	.4	.8	.95	1
9	.4	.8	.95	1
10	.4	.8	.95	1
11	.4	.8	.95	1
12	.4	.8	.95	1
13	.4	.8	.95	1;

TABLE

WORK(I,ITV) Maximum work time per interval in hours

	1	2	3	4
1	3.0	5.0	7.0	10.0
2	2.75	5.5	7.7	11.0
3	2.75	5.5	7.7	11.0
4	5.1	8.5	11.9	17.0
5	2.1	3.5	4.9	7.0
6	2.5	5.0	7.0	10.0
7	3.0	5.0	7.0	10.0
8	2.25	4.5	6.3	9.0
9	2.75	5.5	7.7	11.0
10	2.75	5.5	7.7	11.0
11	2.5	5.0	7.0	10.0
12	2.75	5.5	7.7	11.0
13	2.75	5.5	7.7	11.0;

TABLE

DIST(I,J) Distance from one position to another in Km

	1	2	3	4	5
1	0	1.5	6.0	2.5	6.5
2		0	5.0	1.8	6.0
3			0	2.0	2.5
4				0	4.0
5					0
O	18.0	17.0	14.5	15.5	14.5

```
  +     6      7      8       9
 1     9.0   11.0    9.5    11.0
 2     8.0   10.0    8.5     9.0
 3     5.0    7.0    5.5     6.5
 4     7.0    9.0    7.5     7.5
 5     3.0    3.5    6.0     7.0
 6      0     2.0    5.5     7.5
 7             0     7.0     8.5
 8                    0       0
 9                            0
 O    13.0   14.5    7.0     6.0

  +    10     11      12      13      T
 1    7.0   14.0    18.0    21.0    33.0
 2    6.0   12.5    17.0    20.0    32.0
 3    7.0   10.0    14.5    17.5    29.5
 4    5.0   11.0    15.5    18.5    30.5
 5    7.0   10.0    14.5    17.5    29.5
 6    9.0    8.5    13.0    16.0    28.0
 7   10.5   10.0    14.5    17.5    29.5
 8    6.5    4.0     7.0    10.0    22.0
 9    5.0    3.0     6.0     9.0    21.0
10     0     7.5    11.0    14.0    26.0
11            0      4.0     7.0    19.0
12                    0      3.0    15.0
13                            0     18.0
 O   11.0   14.0      0      3.0      0   ;

TABLE

  RATE(I,J)  Rate of movement from i to j in Kmh

        1      2      3      4      5
 1      0     25     20     25     25
 2             0     20     25     25
 3                    0     20     20
 4                           0     25
 5                                  0
 O     25     25     22     25     25

  +     6      7       8       9
 1     20     20      20      25
 2     20     20      20      25
 3     15     15      15      20
 4     20     20      20      25
 5     20     20      20      25
 6      0     15      15      20
 7             0      15      20
 8                     0      10
 9                             0
 O     22     22      20      25
```

```
   +      10     11     12     13      T
  1       25     20     20     20     25
  2       25     20     20     20     25
  3       20     15     15     15     22
  4       25     20     20     20     25
  5       25     20     20     20     25
  6       20     15     20     20     22
  7       20     15     20     20     22
  8       20     20     22     22     22
  9       25     25     25     25     25
 10        0     20     25     25     25
 11               0     20     20     22
 12                      0     20     22
 13                             0     22
  O       25     20     20     20      0;

TABLE

   COMPAT(K,I)    Asset k can be used at position i

               1  2  3  4  5  6  7  8  9  10  11  12  13
 ACE1-ACE2     1  1  1  1  1  1  1  1  1   1   1   1   1
 ACE3-ACE4     1  1  1  1  1  1  1  1  1   1   1   1   1
 ACE5-ACE6     1  1  1  1  1  1  1  1  1   1   1   1   1
 CEV1-CEV2     1  1  1  1  1  1  1  1  1   1   1   1   1;


 SCALAR   FINISH   Time to arrive at destination      /78/;
```

3. GAMS FORMULATION

```
$TITLE                    ENGINEER MODEL

*-----------------------------------------------------------
*                    GAMS OPTIONS SETTINGS
*-----------------------------------------------------------
$OFFUPPER OFFSYMXREF OFFSYMLIST
 OPTIONS LIMCOL =000, LIMROW = 000, SOLPRINT = OFF
  RESLIM = 19999,ITERLIM = 2500000,OPTCR = 0.1,WORK = 100000;

*-----------------------------------------------------------
*                         GIVEN DATA
*-----------------------------------------------------------
$INCLUDE MINEWIRE DATA A
```

```
*----------------------------------------------------------------
*                       DERIVED DATA
*----------------------------------------------------------------
DIST(I,J) $ (ORD(I) GT ORD(J) AND
             ORD(I) LE CARD(I)-2) = DIST(J,I);
RATE(I,J) $ (ORD(I) GT ORD(J) AND
             ORD(I) LE CARD(I)-2) = RATE(J,I);

PARAMETER PDIST(IP,J);
  PDIST(IP,J)=SUM(I $ (ORD(IP) EQ ORD(I) AND
                       ORD(I) LE CARD(IP)  ), DIST(I,J));

PARAMETER PRATE(IP,J);
  PRATE(IP,J) = SUM(I $ (ORD(IP) EQ ORD(I) AND
                         ORD(I) LE CARD(IP)   ), RATE(I,J));

PARAMETER PCOMP(K,IP);
   PCOMP(K,IP) = SUM(I $ (ORD(IP) EQ ORD(I) AND
                          ORD(I) LE CARD(IP)  ), COMPAT(K,I));
*----------------------------------------------------------------
*                    DECISION VARIABLES
*----------------------------------------------------------------
POSITIVE VARIABLES

  S(K,V)     Arrival time at node V
  X(K,I)     Amount of time for asset k to spend at position i
  W(I,ITV)   Work for interval itv ;

   X.UP(K,I) = D(I);
   S.UP(K,IP) = T(IP) - (TRUNC(T(IP)*MNTPRD(K)/24))*MNTDUR(K);
   S.UP(K,'T') =
          FINISH-(TRUNC(FINISH*MNTPRD(K)/24))*MNTDUR(K);
   W.UP(I,ITV) = WORK(I,ITV);

BINARY VARIABLE

  Z(K,V,V)         Flow on arc from one node to another node

VARIABLE

  RETURN           Total value of engineer plan
  TIME             Find the shortest path;
```

```
*-----------------------------------------------------------------
*                        FORMULATION
*-----------------------------------------------------------------
EQUATIONS

 OBJ              Total value of engineer plan
 ORIGIN(K)        Ensure balance of flow at origin
 POSBAL(K,I)      Ensure balance at each position
 ARTBAL(K,IP)     Ensure balance of flow at each artificial node
 DEST(K)          Ensure balance of flow at destination
 DUALA(K,I,J)     Calculate asset k arrival time at first task i
 DUALB(K,I,IP)    Calculate asset k departure time from task i
 DUALC(K,IP,I)    Calculate asset k arrival for following task i
 TOTWRK(I)        Total amount of work done at i
 BEGWRK(I)        Indicate work begun at position i
 INTWRK(I,ITV)    Work done in interval itv at position i
 ONEAST(I)        Ensure only one asset visits position i
 FINT(K,IP,I)     Calculate arrival time at T
 EQU(K,I)         Asset k does work at i only if k visits i;


 OBJ..
   RETURN =E= SUM(I, P(I) * SUM(ITV $ (ORD(ITV) GE 2),
((LEVEL(I,ITV)-LEVEL(I,ITV-1))/(WORK(I,ITV)-WORK(I,ITV-1))*
     W(I,ITV) + LEVEL(I,'1') * W(I,'1'))));

 ORIGIN(K)..
   SUM(I $ (ORD(I) LE CARD(IP) AND COMPAT(K,I) EQ 1),
         Z(K,'O',I)) =E= 1;

 POSBAL(K,I) $ (ORD(I) LE CARD(IP) AND COMPAT(K,I) eq 1)..
   Z(K,'O',I) + SUM(IP $ (ORD(I) NE ORD(IP) AND
                  PCOMP(K,IP) EQ 1         ), Z(K,IP,I))
            =E= SUM(IP $ (ORD(I) EQ ORD(IP) AND
                 PCOMP(K,IP) EQ 1          ), Z(K,I,IP));

 ARTBAL(K,IP) $ (PCOMP(K,IP) EQ 1)..
   SUM(I $ (ORD(I) EQ ORD(IP) AND COMPAT(K,I) EQ 1),Z(K,I,IP))
     =E= Z(K,IP,'T') + SUM(I $ (ORD(I) NE ORD(IP) AND
                               ORD(I) LE CARD(IP) AND
                               COMPAT(K,I) EQ 1     ),
                                Z(K,IP,I));

 DEST(K)..
   SUM(IP $ (PCOMP(K,IP) EQ 1), Z(K,IP,'T')) =E= 1;

 DUALA(K,'O',I) $ (COMPAT(K,I) EQ 1 AND ORD(I) LE CARD(IP))..
   S(K,'O') + DIST('O',I)/RATE('O',I) =L=
            S(K,I) + (1-Z(K,'O',I)) * 500;
```

```
DUALB(K,I,IP) $ (COMPAT(K,I) EQ 1 AND ORD(I) EQ ORD(IP))..
  S(K,I) + X(K,I) =L= S(K,IP) + (1-Z(K,I,IP)) * 500;

DUALC(K,IP,I) $ (COMPAT(K,I) EQ 1 AND PCOMP(K,IP) EQ 1
                                   AND ORD(I) NE ORD(IP) AND
                                   ORD(I) NE (CARD(IP) + 1))..
       S(K,IP)+PDIST(IP,I)/PRATE(IP,I) =L=
                                   S(K,I) + (1-Z(K,IP,I)) * 500;

TOTWRK(I) $ (ORD(I) LE CARD(IP))..
  SUM(ITV, W(I,ITV)) =E= SUM(K $ (COMPAT(K,I) EQ 1),
                                       (X(K,I)*FACTOR(K)));

BEGWRK(I) $ (ORD(I) LE CARD(IP))..
  W(I,'1') =E= WORK(I,'1') * SUM((K,IP) $ (COMPAT(K,I) EQ 1
                                       AND ORD(I) EQ ORD(IP)   ),
                                       Z(K,I,IP));

INTWRK(I,ITV) $ (ORD(I) LE CARD(IP))..
  W(I,ITV) =L= (WORK(I,ITV) - WORK(I,ITV - 1)) *
         SUM((K,IP) $ (PCOMP(K,IP) EQ 1 AND
                       ORD(I) EQ ORD(IP)  ), Z(K,I,IP));

ONEAST(I)..
  SUM((K,IP) $ (ORD(I) EQ ORD(IP) AND COMPAT(K,I) EQ 1),
         Z(K,I,IP)) =L= 1;

FINT(K,IP,'T') $ (PCOMP(K,IP) EQ 1)..
  S(K,IP) + PDIST(IP,'T')/PRATE(IP,'T') =L=
       S(K,'T')+ 500*(1-Z(K,IP,'T'));


EQU(K,I) $ (COMPAT(K,I) EQ 1)..
  FACTOR(K)*X(K,I) =L= SUM(IP $ (ORD(I) EQ ORD(IP)),
        Z(K,I,IP))*D(I);


MODEL ENGINEER /ALL/;

SOLVE ENGINEER USING MIP MAXIMIZING RETURN;


*------------------------------------------------------------
*               ELIMINATE LOITER TIME
*------------------------------------------------------------
*------------------------------------------------------------
*                    DERIVED DATA
*------------------------------------------------------------
X.FX(K,I) = X.L(K,I);
W.FX(I,ITV) = W.L(I,ITV);
```

```
 ALIAS(V,VP);

 Z.FX(K,V,VP) = Z.L(K,V,VP);

 TIME.UP = SUM(K, S.L(K,'T'));

 PARAMETER WD(K,I);
 WD(K,I) = X.L(K,I);

*------------------------------------------------------------
*                       FORMULATION
*------------------------------------------------------------
EQUATIONS

  OBJ2               Minimize loiter time
  ORIGIN2(K)         Ensure balance of flow at origin
  DEST2(K)           Ensure balance of flow at destination;

  OBJ2..
        TIME =E= sum((k,I,IP) $ (PCOMP(K,IP) EQ 1  AND
                                 ORD(I) EQ ORD(IP) AND
                                 COMPAT(K,I) EQ 1  AND
                                 WD(K,I) GT 0          ),
                                 S(K,IP) - S(K,I));


  ORIGIN2(K)..
        SUM(I $ (ORD(I) LE CARD(IP) AND COMPAT(K,I) EQ 1),
            Z(K,'O',I)) =L= 1;

  DEST2(K)..
        SUM(IP $ (PCOMP(K,IP) EQ 1), Z(K,IP,'T')) =L= 1;

  MODEL ENGINEER2 /OBJ2, DUALA, DUALB,DUALC,FINT/;

  SOLVE ENGINEER2 USING RMIP MINIMIZING TIME;

  OPTION Z:2:1:1;

  PARAMETER ASCHED(K,I,*), ROLLUP(I,K,*);

  ASCHED(K,I,'ARRTIME') $ (X.L(K,I) GT 0)= S.L(K,I) *
           SUM(IP $ (ORD(I) EQ ORD(IP)
                  AND COMPAT(K,I) EQ 1
                  AND PCOMP(K,IP) EQ 1),
                  Z.L(K,I,IP)        )
            + TRUNC((S.L(K,I) $ (X.L(K,I) GT 0))
                         *MNTPRD(K)/24)*MNTDUR(K);

 ASCHED(K,I,'EFFORT') = X.L(K,I);
```

```
ASCHED(K,I,'DEPART') = SUM(IP $ (ORD(I) EQ ORD(IP)
                           AND X.L(K,I) GT 0), S.L(K,IP)) +
           TRUNC(SUM(IP $ (ORD(I) EQ ORD(IP)
                          AND X.L(K,I) GT 0), (S.L(K,IP)))
                       *MNTPRD(K)/24) * MNTDUR(K);

ASCHED(K,I,'DUE') = SUM(IP $ (ORD(I) EQ ORD(IP)
                        AND X.L(K,I) GT 0), T(IP));

ASCHED(K,I,'MAINT') $ (X.L(K,I) GT 0) =
      TRUNC(ASCHED(K,I,'DEPART') - ASCHED(K,I,'ARRTIME') -
                   X.L(K,I)+10E-8);

ASCHED(K,I,'PRIORITY') = P(I) $ (X.L(K,I) GT 0);
ASCHED(K,I,'DESTTIME') $ (X.L(K,I) GT 0)= S.L(K,'T')
           + TRUNC((S.L(K,'T') * MNTPRD(K)/24))*MNTDUR(K);
ROLLUP(I,K,'ARRTIME')= ASCHED(K,I,'ARRTIME');
ROLLUP(I,K,'EFFORT') = ASCHED(K,I,'EFFORT');
ROLLUP(I,K,'DEPART') = ASCHED(K,I,'DEPART');
ROLLUP(I,K,'DUE') = ASCHED(K,I,'DUE');
ROLLUP(I,K,'MAINT') = ASCHED(K,I,'MAINT');
ROLLUP(I,K,'DESTTIME') = ASCHED(K,I,'DESTTIME');
ROLLUP(I,K,'PRIORITY') = ASCHED(K,I,'PRIORITY');

OPTION ASCHED:2:1:1;
OPTION ROLLUP:2:2:1;
DISPLAY ASCHED, ROLLUP;
```

APPENDIX C: SAMPLE SCENARIO

1. GENERAL

The chosen scenario occurs at the National Training Center, Fort Irwin, California [Ref 12]. The task force mission is to defend against a motorized rifle regiment attack. During the development of the scenario, it was determined that a number of the maneuver units would execute movements 'on order'. In order to generate the data required for the Maneuver model, these 'on order' movements were given associated times consistent with Soviet doctrine as given in FM 100-2-1, and with estimates of how long each engagement would last.

2. SCENARIO

Figure C.1 is the general overlay for the operation. The task force is tank-heavy with two tank companies, one tank-heavy team, and one mech-heavy team. Only the maneuver units and engineers were considered for this scenario.

The probable enemy avenue of approach (AA) is indicated in Figure C.1. The regiment is in march formation as the division lead element.

The task force course of action requires the two tank companies to engage the enemy forces from battle positions (BP's) 101 and 102. These companies will engage the enemy up through the lead elements of the regimental main body in

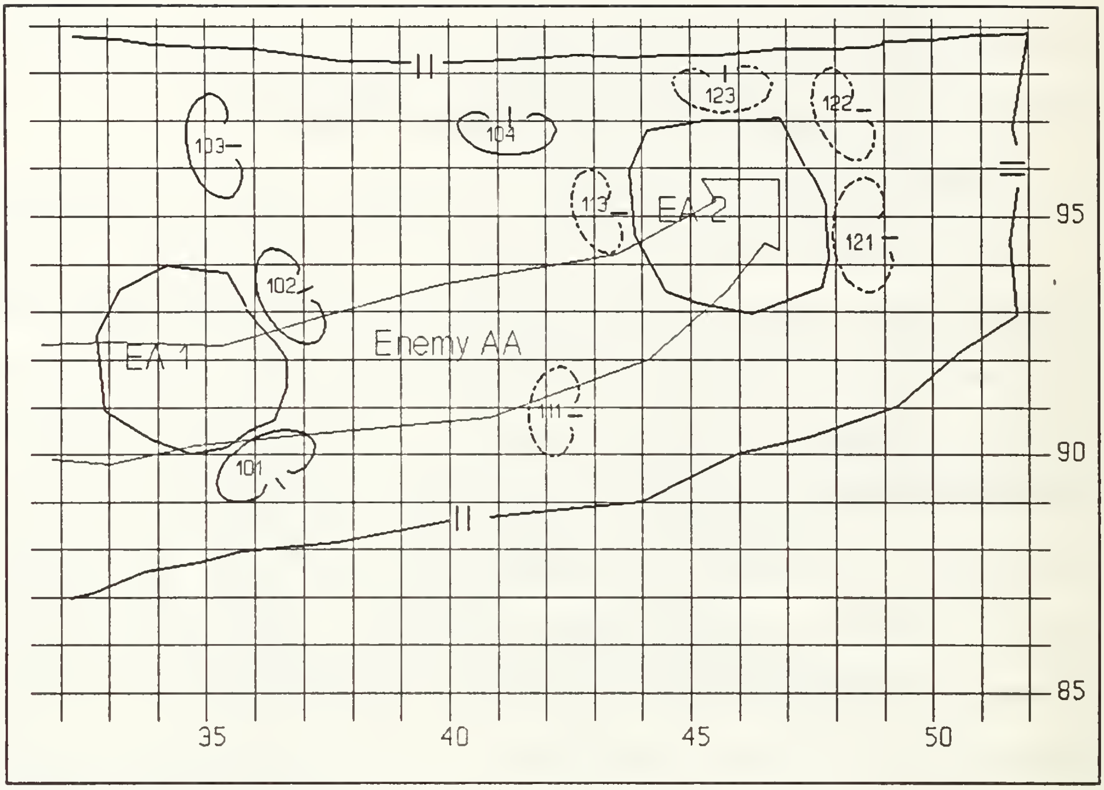

Figure C.1 Sample Scenario Overlay

engagement area (EA) 1. The tank-heavy team blocks a potential enemy avenue of approach to the north from BP 103. On order, the tank company in BP 102 withdraws to BP 122 for the follow-on engagement in EA 2, while the tank heavy team and other tank company withdraw to BP's 113 and 111, respectively, in order to fight a delaying action. The mech-heavy team engages the enemy's flank security elements from BP 104. Once again, on order, the tank heavy team and tank company withdraw to BP's 123 and 121 respectively for the final engagement.

The engineer tasks required for the scenario include 'digging in' all vehicles, and emplacement of mine, wire and anti-tank ditch obstacles throughout the depth of the battlefield. Initially, the task force has a heavy division engineer company under operational control. Only the six Armored Combat Earthmovers (ACE), two Combat Engineer Vehicles (CEV), three engineer platoons, and GEMSS assets of this company were used as input into the model. Subsequently, as stated in Chapter IV, the number of available assets was decreased for illustrating flexibility. The activities in the other battlefield operating systems are assumed to be fully synchronized i.e., mines and wire are available when required by the engineers.

LIST OF REFERENCES

1. FM 100-5, Operations, Department of the Army, May 1986.

2. FM 71-2, The Tank and Mechanized Infantry Battalion Task Force, Department of the Army, September 1988.

3. FM 101-5, Staff Organization and Operations, Department of the Army, May 1984.

4. Long, Clyde C., MAJ, USA, Synchronization of Combat Power at the Task Force Level: Defining a Planning Methodology, M.A. Thesis, U.S. Army Command and General Staff College, Fort Leavenworth, Kansas, June 1989.

5. Winston, Wayne L., Operations Research: Applications and Algorithms, Duxbury Press, 1987.

6. FM 100-2-1, The Soviet Army: Operations and Tactics, Department of the Army, July 1984.

7. FM 5-34, Engineer Field Data, Department of the Army, September 1987.

8. Brooke, A., Kendrick, D., and Meeraus, A., GAMS: A User's Guide, Scientific Press, 1988.

9. Singhal, J., Marsten, R., and Morin, T., Fixed Order Branch-and-Bound Methods for Mixed-Integer Programming: The Zoom System, University of Arizona, December 1987.

10. Tactical Commanders' Development Course, Battle Book, U.S. Army Command and General Staff College, Fort Leavenworth, Kansas.

11. IBM, International Business Machines Corporation, Mathematical Programming System-Extended (MPSX) and Generalized Upper Bounding (GUB) Program Description, IBM Manual SH20-0968-1, White Plains, New York, 1972.

12. Map Sheet, "Fort Irwin Military Installation Map South", Edition 2-DMA, Series V7955, Sheet Fort Irwin MIM SOUTH, Defense Mapping Agency, 1984.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Professor Siriphong Lawphongpanich, Code OR/Lp Department of Operations Research Naval Postgraduate School Monterey, California 93943-5000	2
4.	U.S. Total Army Personnel Command ATTN: TAPC-PLA (CPT Morey) Alexandria, Virgina 22332-0400	2
5.	Director U.S. Army TRADOC Analysis Command-Monterey ATTN: ATRC-RDM (CPT Dryer) P.O. Box 8692 Monterey, California 93943-0692	3
6.	U.S. Army Command and General Staff College ATTN: ATZL-SWK (LTC Savoie) Fort Leavenworth, Kansas 66027-6900	1
7.	U.S. Army Library ANRAL-RS ATTN: ASDIRS Room 1A518, The Pentagon Washington, D.C. 20310	1
8.	U.S. Army Combined Arms Research Library (CARL) ATTN: ATZL-SWS-L Fort Leavenworth, Kansas 66027	1
9.	Director CAC Future Battle Lab ATTN: ATZL-CDC-B Fort Leavenworth, Kansas 66027	1

Thesis
M82245 Morey
c.1 Optimization models
for synchronization
planning.